Project 5 - Can we predict whether a Hotel Booking will be canceled?

Ryan Chen 113200236

The Hotel Booking Dataset (Kaggle)

- Hotel Booking records for city and resort hotels
- Data collected from July 1, 2015 to August 31, 2017
- Dataset Size
  - 119390 rows
  - 36 columns

An Example Booking

```
hotel                                    Resort Hotel
is_canceled                                         0
lead_time                                          14
arrival_date_year                                2015
arrival_date_month                               July
arrival_date_week_number                           27
arrival_date_day_of_month                           1
stays_in_weekend_nights                             0
stays_in_week_nights                                2
adults                                              2
children                                          0.0
babies                                              0
meal                                               BB
country                                           GBR
market_segment                              Online TA
distribution_channel                            TA/TO
is_repeated_guest                                   0
previous_cancellations                              0
previous_bookings_not_canceled                      0
reserved_room_type                                  A
assigned_room_type                                  A
booking_changes                                     0
deposit_type                               No Deposit
```

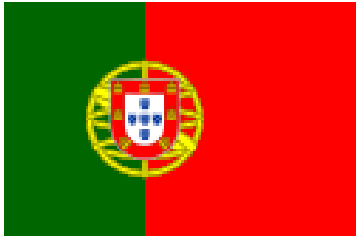
```
agent                                           240.0
company                                           NaN
days_in_waiting_list                                0
customer_type                               Transient
adr                                              98.0
required_car_parking_spaces                         0
total_of_special_requests                           1
reservation_status                          Check-Out
reservation_status_date                    2015-07-03
name                                 Jasmine Fletcher
email                         JFletcher43@xfinity.com
phone-number                             190-271-6743
credit_card                          ************9263
Name: 5, dtype: object
```

Exploratory Data Analysis
(Pandas, Numpy, Matplotlib)

Which country saw the most hotel bookings?

df.country.value_counts()

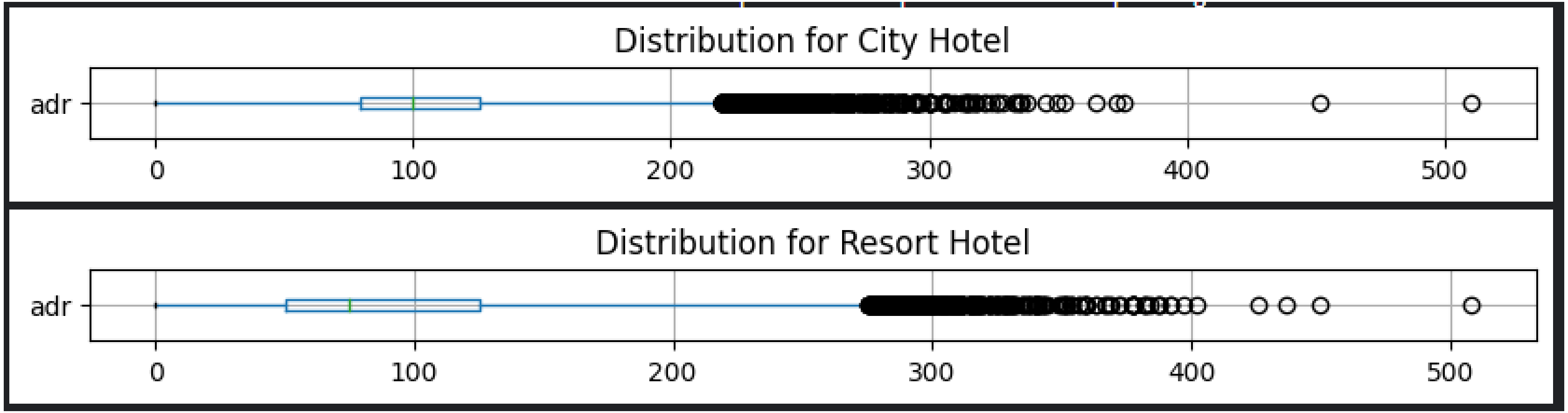
```
country
PRT     48590
GBR     12129
FRA     10415
ESP      8568
DEU      7287
         ...
DJI         1
BWA         1
HND         1
VGB         1
NAM         1
Name: count, Length: 177, dtype: int64
```



PRT = Portugal accounts for 40.7 % of all bookings in the data

What is the distribution like for both hotels with respect to the price of a room per night?

column of interest = Average Daily Rate (adr)



Which months are most busy for both hotels?

```
df[df.hotel==h_type]
.arrival_date_month
.value_counts().
head(3).index
```

City Hotel

- May
- July
- August

Resort Hotel

- April
- July
- August

Which months see the most expensive per night costs?

```
df[df.hotel==h_type]
.groupby('arrival_date_month')
['adr']
.mean().sort_values()
.tail(3)
```

```
The most expensive per night costs for City Hotel are:
arrival_date_month
August     114.68
June       119.07
May        121.64
Name: adr, dtype: float64
```

```
The most expensive per night costs for Resort Hotel are:
arrival_date_month
June       110.44
July       155.18
August     186.79
Name: adr, dtype: float64
```

Which months see the most cancellations for both hotel types?

df[df.hotel == h_type]

.groupby('arrival_date_month')

['is_canceled']

.sum().head()

```
For City Hotel , these months gave most cancellations
arrival_date_month
April        3465
August       3602
December     1740
February     1901
January      1482
Name: is_canceled, dtype: int64

For Resort Hotel , these months gave most cancellations
arrival_date_month
April        1059
August       1637
December      631
February      795
January       325
Name: is_canceled, dtype: int64
```

Examine distributions of bookings vs market segment

df.market_segment.value_counts()

```
Examine distributions of bookings vs market segment.
market_segment
Online TA        56477
Offline TA/TO    24218
Groups           19810
Direct           12606
Corporate         5295
Complementary      743
Aviation           237
Undefined            2
Name: count, dtype: int64
```

Which room type was most commonly booked? Most commonly cancelled?

```
reserved_room_type
A    85992
D    19201
E     6535
F     2897
G     2094
B     1118
C      932
H      601
P       12
L        6
Name: count, dtype: int64
```

```
reserved_room_type
A    33629
D     6102
E     1914
F      880
G      763
B      368
C      308
H      245
P       12
L        2
Name: is_canceled, dtype: int64
```

What percentage of the data recorded cancellations for each hotel?

df[df.hotel==h_type]

.is_canceled.sum() *100

/ df[df.hotel==h_type].shape[0]

```
is_canceled
0    75166
1    44224
Name: count, dtype: int64
```

Result

City Hotel bookings

- 41.73 % of bookings cancelled

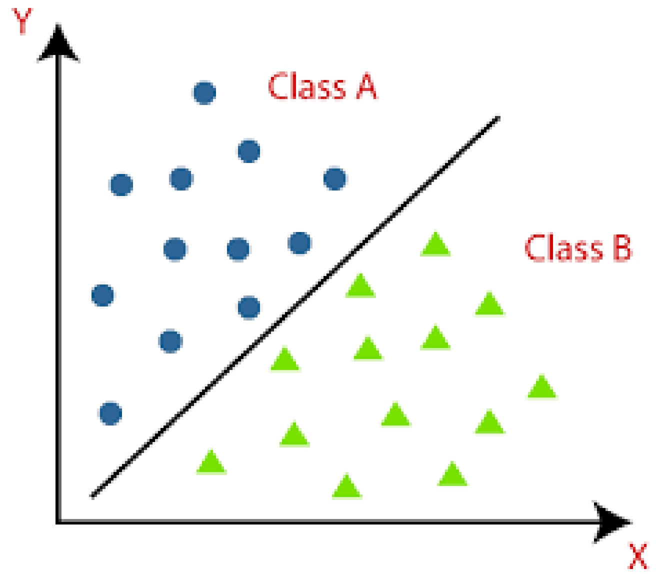
Resort Hotel bookings

- 27.76 % of bookings cancelled

Machine Learning

What models to consider?

- Hotel Booking Cancellation is a classification task!
- A booking can be
  - canceled (1)
  - not canceled (0)



Data Cleaning (1)

- We have an overwhelming amount of features (36)
- Must eliminate some columns, initial dropping shown

```
['lead_time','arrival_date_year','arrival_date_week_number',
 'arrival_date_day_of_month','previous_bookings_not_canceled',
 'booking_changes','deposit_type','agent','company',
 'required_car_parking_spaces','reservation_status',
 'reservation_status_date','name','email','phone-number','credit_card']
```

Data Cleaning (2) - numerical cols

```
for col in ['adr', 'days_in_waiting_list']:
    df[col] = np.log(df[col] + 0.000001)
```

is_canceled	Mean: 0.37	STD: 0.48	Min: 0.0	Median: 0.0	Max: 1.0
stays_in_weekend_nights	Mean: 0.93	STD: 1.0	Min: 0.0	Median: 1.0	Max: 19.0
stays_in_week_nights	Mean: 2.5	STD: 1.91	Min: 0.0	Median: 2.0	Max: 50.0
adults	Mean: 1.86	STD: 0.58	Min: 0.0	Median: 2.0	Max: 55.0
children	Mean: 0.1	STD: 0.4	Min: 0.0	Median: 0.0	Max: 10.0
babies	Mean: 0.01	STD: 0.1	Min: 0.0	Median: 0.0	Max: 10.0
is_repeated_guest	Mean: 0.03	STD: 0.18	Min: 0.0	Median: 0.0	Max: 1.0
previous_cancellations	Mean: 0.09	STD: 0.84	Min: 0.0	Median: 0.0	Max: 26.0
days_in_waiting_list	Mean: 2.32	STD: 17.59	Min: 0.0	Median: 0.0	Max: 391.0
adr	Mean: 101.79	STD: 48.15	Min: 0.0	Median: 94.575	Max: 510.0
total_of_special_requests	Mean: 0.57	STD: 0.79	Min: 0.0	Median: 0.0	Max: 5.0

Data Cleaning (3) - categorical vars

```
Col hotel has ['City Hotel' 'Resort Hotel'] unique values
Col arrival_date_month has ['April' 'August' 'December' 'February' 'January' 'July' 'June' 'March'
 'May' 'November' 'October' 'September'] unique values
Col meal has ['BB' 'FB' 'HB' 'SC' 'Undefined'] unique values
Col country has ['ABW' 'AGO' 'AIA' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ASM' 'ATA' 'ATF' 'AUS'
 'AUT' 'AZE' 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR'
 'BOL' 'BRA' 'BRB' 'BWA' 'CAF' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'CN' 'COL'
 'COM' 'CPV' 'CRI' 'CUB' 'CYM' 'CYP' 'CZE' 'DEU' 'DJI' 'DMA' 'DNK' 'DOM'
 'DZA' 'ECU' 'EGY' 'ESP' 'EST' 'ETH' 'FIN' 'FJI' 'FRA' 'FRO' 'GAB' 'GBR'
 'GEO' 'GGY' 'GHA' 'GIB' 'GLP' 'GNB' 'GRC' 'GTM' 'GUY' 'HKG' 'HND' 'HRV'
 'HUN' 'IDN' 'IMN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JEY'
 'JOR' 'JPN' 'KAZ' 'KEN' 'KHM' 'KIR' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBY'
 'LCA' 'LIE' 'LKA' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MCO' 'MDG' 'MDV' 'MEX'
 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'MYT' 'NAM'
 'NCL' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL'
 'PLW' 'POL' 'PRI' 'PRT' 'PRY' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAU' 'SDN'
 'SEN' 'SGP' 'SLE' 'SLV' 'SMR' 'SRB' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SYC'
 'SYR' 'TGO' 'THA' 'TJK' 'TMP' 'TUN' 'TUR' 'TWN' 'TZA' 'UGA' 'UKR' 'UMI'
 'URY' 'USA' 'UZB' 'VEN' 'VGB' 'VNM' 'ZAF' 'ZMB' 'ZWE'] unique values
Col market_segment has ['Aviation' 'Complementary' 'Corporate' 'Direct' 'Groups' 'Offline TA/TO'
 'Online TA'] unique values
Col distribution_channel has ['Corporate' 'Direct' 'GDS' 'TA/TO' 'Undefined'] unique values
Col reserved_room_type has ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'L' 'P'] unique values
Col assigned_room_type has ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'K' 'L' 'P'] unique values
Col customer_type has ['Contract' 'Group' 'Transient' 'Transient-Party'] unique values
```

How to deal with categorical data?

- Many ML models can only accept numerical data
- Solution = use encoding techniques
  - One Hot Encoding
  - Ordinal Encoding

```python
df_encoded = pd.get_dummies(df, columns=['hotel', 'meal', 'market_segment',
                                         'distribution_channel', 'reserved_room_type',
                                         'assigned_room_type', 'customer_type'],
                            drop_first=True)  # drop_first=True to avoid dummy variable trap
```

```python
# ordinal encode the arrival_date_month column
from sklearn.preprocessing import OrdinalEncoder
df = df_encoded
month_order = ['January', 'February', 'March', 'April', 'May', 'June',
               'July', 'August', 'September', 'October', 'November', 'December']
ordinal_encoder = OrdinalEncoder(categories=[month_order])
df['arrival_date_month_encoded'] = ordinal_encoder.fit_transform(df[['arrival_date_month']])
df = df.drop(columns=['arrival_date_month'])
print(df.iloc[5])
```

The resulting columns shown

```
is_canceled                                0
stays_in_weekend_nights                    0
stays_in_week_nights                       2
adults                                     2
children                                 0.0
babies                                     0
is_repeated_guest                          0
previous_cancellations                     0
days_in_waiting_list              -13.815511
adr                                 4.584967
total_of_special_requests                  1
hotel_Resort Hotel                      True
meal_FB                                False
meal_HB                                False
meal_SC                                False
meal_Undefined                         False
market_segment_Complementary           False
market_segment_Corporate               False
market_segment_Direct                  False
market_segment_Groups                  False
market_segment_Offline TA/TO           False
market_segment_Online TA                True
distribution_channel_Direct            False
distribution_channel_GDS               False
distribution_channel_TA/TO              True
distribution_channel_Undefined         False
```

```
reserved_room_type_B                   False
reserved_room_type_C                   False
reserved_room_type_D                   False
reserved_room_type_E                   False
reserved_room_type_F                   False
reserved_room_type_G                   False
reserved_room_type_H                   False
reserved_room_type_L                   False
reserved_room_type_P                   False
assigned_room_type_B                   False
assigned_room_type_C                   False
assigned_room_type_D                   False
assigned_room_type_E                   False
assigned_room_type_F                   False
assigned_room_type_G                   False
assigned_room_type_H                   False
assigned_room_type_I                   False
assigned_room_type_K                   False
assigned_room_type_L                   False
assigned_room_type_P                   False
customer_type_Group                    False
customer_type_Transient                 True
customer_type_Transient-Party          False
arrival_date_month_encoded               6.0
Name: 5, dtype: object
```

Which classifiers should we choose?

1) Logistic Regression
2) Decision Tree
3) Gradient Boosting Classifier

* sklearn is our library of choice here!

But first …. Train Test Split

```python
# split the data into train and test sets
from sklearn.model_selection import train_test_split
X = df.drop(columns=['is_canceled'])
y = df['is_canceled']
assert 'is_canceled' not in X.columns
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42)
```

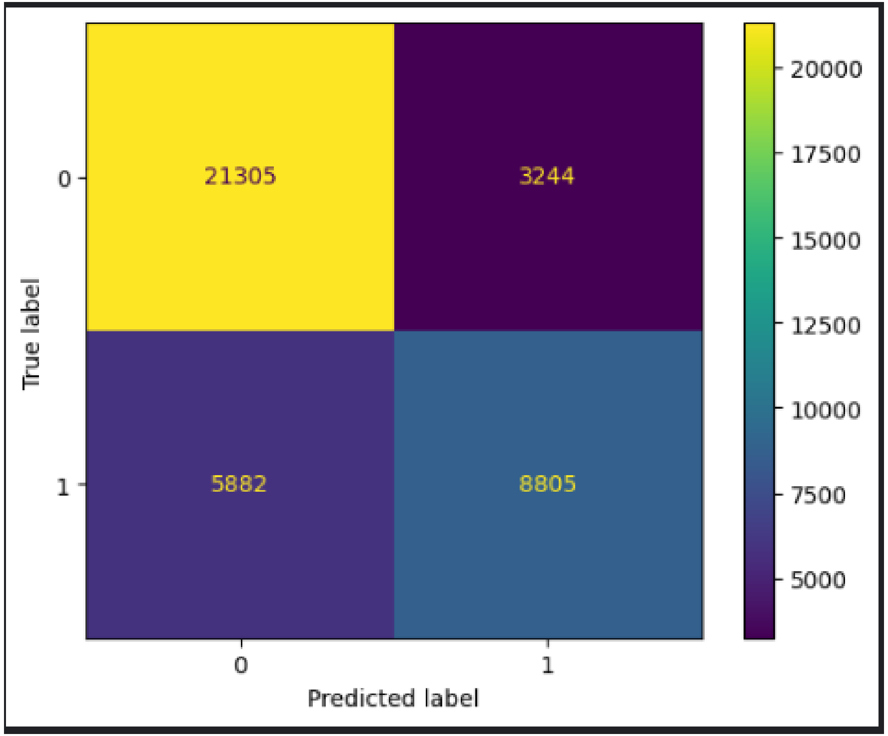
Model Training

```python
# decision tree model
from sklearn import tree
decision_tree = tree.DecisionTreeClassifier()
decision_tree.fit(X_train,y_train)
y_pred = decision_tree.predict(X_test)
```

```python
# Logistic Regression Model
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train, y_train)
y_pred = lr_model.predict(X_test)
print(y_pred)
```

```python
# gradient boosting classifier
from sklearn.ensemble import GradientBoostingClassifier
grad_boost_model = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,
                                              max_depth=1, random_state=0).fit(X_train, y_train)
y_pred = grad_boost_model.predict(X_test)
```

Classification Performance



How to compute these measures?

```
print("The decision tree model results")
acc = metrics.accuracy_score(y_test, y_pred)
print("Accuracy is ",round(acc,2))
precision = metrics.precision_score(y_test, y_pred)
print("Precision is ",round(precision,2))
recall = metrics.recall_score(y_test, y_pred)
print("Recall is ",round(recall,2))
f1_score = metrics.f1_score(y_test, y_pred)
print('F1 score is ',round(f1_score,2))
```

Overall results + Winner!!

Model	Accuracy	Precision	Recall	F1 Score
Log Reg	0.77	0.73	0.6	0.66
Decision Tree	0.79	0.72	0.72	0.72
Grad Boost	0.78	0.75	0.59	0.67

Additional Considerations....

- Additional Feature Reduction
- More models to consider
- Consideration of hyperparameter tuning for each model


THANK YOU
memegenerator.net